Optimizing WebObjects Applications

Session 714

Optimizing WebObjects Applications

Brian Fitzpatrick
Apple Professional Services

Rich Flewelling
Apple Professional Services

Introduction

- Goal—Find and eliminate performance bottlenecks
- Three areas will be addressed
    - Tools to identify trouble spots
    - Types of bottlenecks to look for
    - Techniques which eliminate bottlenecks

What You Will Learn

- When and where to invest time on performance tuning
- How to use WOEventCenter and OptimizeIT to measure performance
- WebObjects and EOF techniques for improving performance
- When it is time to upgrade your hardware

When to Start Tuning?

- Functional stability should come first
- Do not sacrifice good design and maintainability
- Do not optimize without measuring
- Measure, modify, then measure again

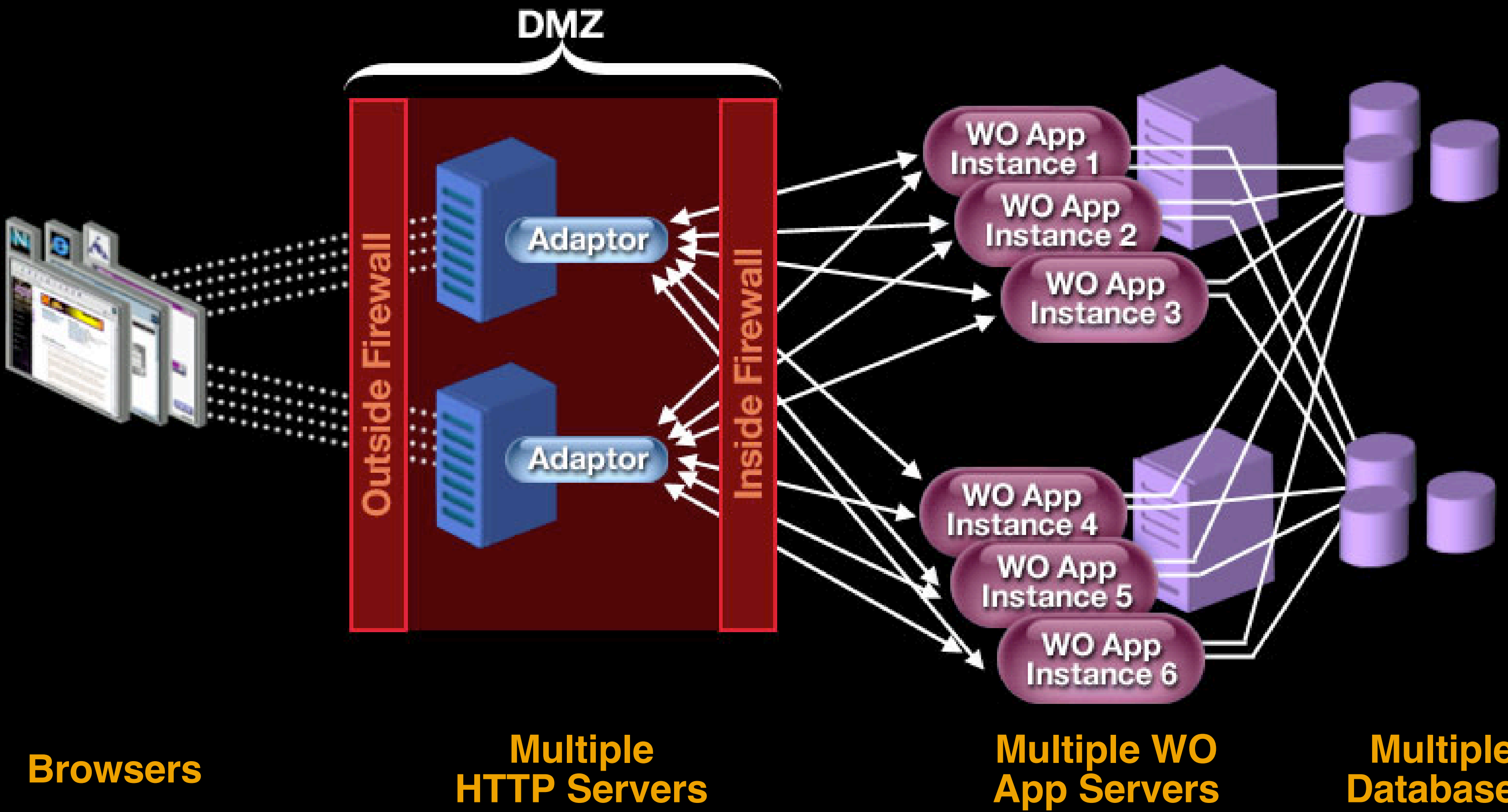
Before You Start

- Make certain your test environment reflects the deployment environment
    - Operating System
    - RAM/CPU/Network
    - Numbers and kinds of servers
- Isolate test environment from users

Before You Start

- Determine which pages are important
- Establish an acceptance criteria
    - All pages should return in under 0.5 seconds
    - The ‘Big Query’ page should return in under 5 seconds
    - Support 400 concurrent users
- Create test cases which exercise important pages

Web App Architectural Overview



Browser/Connectivity

- Performance factors
  - Javascript
  - Cascading Style Sheets
  - Images
  - Connection speed to net
  - Client hardware
  - Browser Plug-ins
  - Network latency

Server Hardware Sizing

- CPU, Memory, Disk I/O, Network
- UNIX utilities
  - ‘top -uw’ (system usage statistics)
  - ‘uptime’ (cpu load for machine)
  - ‘vm_stat 5’ (virtual memory statistics)
- Mac OS X apps
  - CPU Monitor
  - Process Viewer

Demo

Mac OS X Performance Tuning Utilities

Rich Flewelling
Apple Professional Services

Web Server

- Increase the RAM and file cache sizes
- Install the largest, fastest drives possible
- Disable reverse DNS lookups—
  they delay logging
- Install all OS patches

Web Server

- Reduce logging statistics or log to a fast disk subsystem
- Increase the TCP listen backlog if the Web server stops accepting new connections during peak loads
    - Affects how often clients are refused a connection
- Use the latest releases of the http server
- Additional Info
    - **http://httpd.apache.org/docs/misc/perf.html**

Database Server

- Do a system review covering the following:
  - Architecture—Data file locations
  - Hardware sizing
  - Memory Allocations
  - Query tuning
  - Disk I/O, contention
  - Cache data into memory
  - Indexes
  - UNIX kernel configuration

Application-level Tuning

- Process Level
    - Do you have adequate system resources?
        - RAM
        - Number of Application Instances
        - CPU
        - Disk I/O
        - Network bandwidth
    - Adequate numbers of application instances?

Application-level Tuning

Component Level

- Which one of your components is slow?
  - Use WOStats to obtain rough stats on your app
  - Use WOEventCenter to obtain detailed stats on your app
- Is database access slowing you down?
  - Set EOAdaptorDebugEnabled to true

Gathering Data

- View transcription times via WOStats
  http://host/cgi-bin/WebObjects/MyApp.woa/wa/WOStats
- Configure event recording via WOEventSetup
  http://host/cgi-bin/WebObjects/MyApp.woa/wa/WOEventSetup
- View sub-transaction times via the WOEventDisplay page

Interpret Data

- Use WOStats to find pages which do not meet your acceptance criteria
- Use WOEventDisplay to narrow your focus on the source of performance problems
- Evaluate times relative to total time

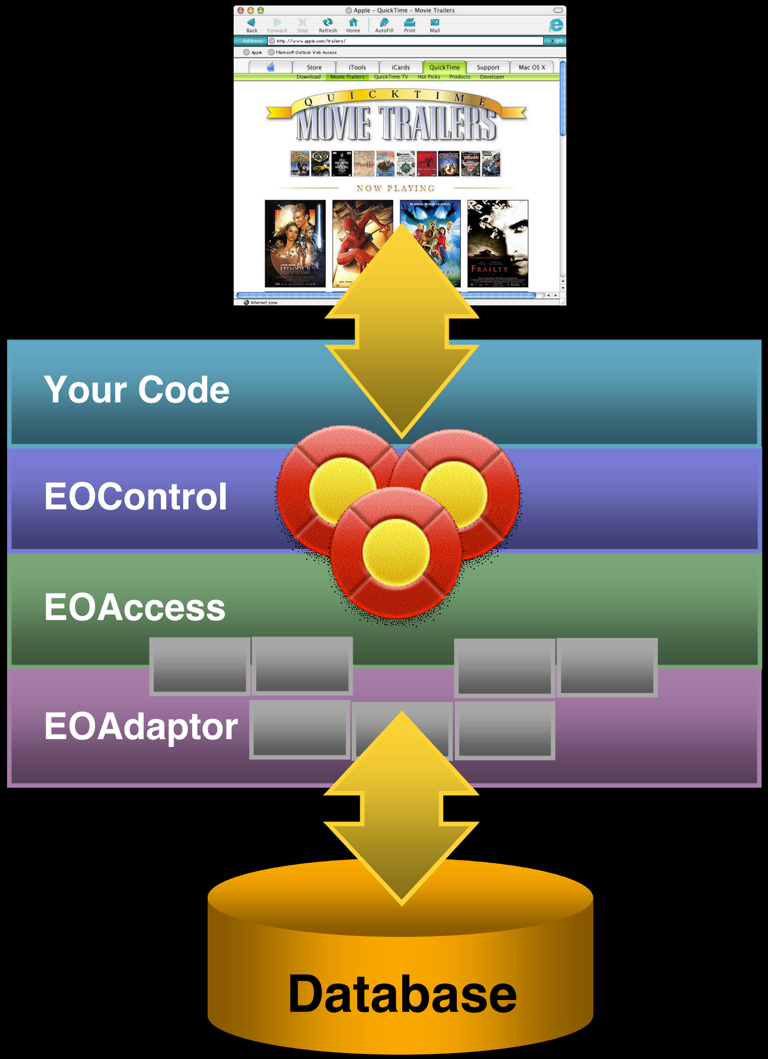
Demo

Using WOStats and WOEventCenter

Brian Fitzpatrick
Apple Professional Services

Slide Depicting App Software Layers

- WO
- Foundation
- Java
- EOF



EOF Performance Improvement

- Reducing trips to the database
- Use batch faulting
- Use prefetching

EOF Performance Improvement

- Using the Shared Editing Context
- Put reference data into the application’s shared editing context
    - E.g., WOPopUps, WOBrowsers containing EOs
- Make your app read-only (very fast)
- Avoid recurring fetches of reference data

EOF Performance Improvement

Reducing the amount of data

- Use raw rows for non-object data and very large fetches
- Use fetch limits
- Do not allow unrestricted searches

EOF Performance

- Managing very large object graphs is slower than small object graphs
  - Use nested editing contexts
  - Or create temporary Editing Contexts and destroy them when no longer used

Managing Very Large Relationships

- Adding or removing an object takes O(n) time and fires all the faults—things are fine for 100 EOs but can be a problem with 100000 EOs
    - Try to minimize the number of adds and removes from a very large to-many relationship

Discarding Objects Cached in Memory

- Use **invalidateAllObjects()** only if you need to refetch the entire object graph
- **invalidateAllObjects()** can be performance intensive
- To refresh a subset of EOs
    - Create an EOFetchSpecification with **setRefreshedRefetchedObjects (true)** in order to refresh the objects fetched
    - Warning: **setRefreshesRefetchedObjects(true)** does not refresh all EOs that are referenced unless they are in the fetched set

Discarding Objects Cached in Memory

- When creating a new Editing Context, use a small **setDefaultFetchTimestampLag()**
  - Default value is one hour
  - Use a smaller value to periodically refresh the snapshots in your object graph as an alternative to **invalidateAllObjects()**

Demo

Using EOF's Tuning Parameters

Rich Flewelling
Apple Professional Services

Application-level Tuning

- Bad Algorithms
- Excessive memory demands
- Memory Leaks
    - Use Optimizeit to look inside the JVM at your running application

Improving Java Performance

- Tune the JVM heap size
- Test with different deployment environments

Tune the JVM Heap Size

- The JVM heap size plays an important role in overall Java application performance
- Minimum heap size = maximum heap size
    - May help with your throughput performance

Tune the JVM Heap Size

- WebObjects 4.5 command line settings **NSJavaMinHeapSize** and **NSJavaMaxHeapSize** now obsolete in WO5
- WO5.1 defaults—Java min heap size=32 MB, Java max heap size=64 MB
- Use command line options: **-Xms#m** (for min heap size) or **-Xmx#m** (for max heap size) where # is the number of MB of RAM to allocate

Deployment Environment

- Case 1—Deployment on a particular machine or a bank of identical machines
- Case 2—Deployment on a wide array of machines with different RAM settings

Deployment: Identical Machines

- Tailor RAM usage to that machine
- Examine the machine with **top** or **vm_stat** to see how much real memory will be available before your application is launched
- Configure the JVM to use most of what is left over

Deployment: Various Machines

- Set your minimum heap size (-Xms#m) to available RAM on the worst machine
- Set your max heap size (-Xmx#m) to available RAM on your best machine
- Test and tune further

Setting References to Null

- Set your object references to null when no longer used
- Example

  If you create your own editing context, call:

  ```
  myEC = null;
  ```

  to nullify the reference

Be Aware of Finalize()

- Calling your own **finalize()** methods can incur a performance penalty because of non-deterministic behavior of garbage collection
- Avoid invoking explicitly **System.gc()** or **runFinalization()** to reduce overhead
- If you must implement **finalize()**, remember to call **super()**

Demo

Using Borland’s Optimizeit with WebObjects

Brian Fitzpatrick
Apple Professional Services

WebObjects Beta

- To be considered for the beta

  Appleseed.apple.com/webobjects

WebObjects Lab

- Located downstairs in Room L
- Lab hours

Monday	12:00pm–6:00pm
Tuesday	9:00am–2:00pm
Wednesday	9:00am–6:00pm
Thursday	9:00am–6:00pm
Friday	9:00am–6:00pm

Roadmap

FF013 WebObjects	Room A1 **Fri., 3:30pm**

Who to Contact

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

Bob Fraser
WebObjects Product Manager
webobjects@apple.com

Apple Professional Services (Training, Support, Consulting)
(800) 848-6398
services@apple.com

http://developer.apple.com/wwdc2002/urls.html

For More Information

- WebObjects Developer Documentation
  http://developer.apple.com/techpubs/webobjects
- Apple Professional Services Technical Support
  www.apple.com/services/technicalsupport
- Other places
  www.apple.com/webobjects
  developer.apple.com/webobjects
  www.apple.com/services
  www.info.apple.com/webobjects
- Subscribe to
  webobjects-announce@apple.com

How to Access Documentation

- Most up-to-date: PDF and HTML

  http://developer.apple.com/techpubs/webobjects
- Hardcopy print-on-demand

  Vervante.com under Related Resources
- Product CD

  Documents folder and installed in
  /Developer/Documentation/WebObjects
- In the box (localized)

  Installation Guides, What's New, WebObjects Overview, Java Client Desktop Applications,
  Discovering WebObjects for HTML
- Check ADC News for latest updates

  http://developer.apple.com/devnews

Q&A

Toni Trujillo Vian
Director, WebObjects Engineering
webobjects@apple.com

http://developer.apple.com/wwdc2002/urls.html

WWDC2002

WWDC2002

WWDC2002